

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-185389

(43)Date of publication of application : 16.07.1996

(51)Int.Cl.

G06F 17/14
G06T 1/00
G10L 9/16
H03H 7/30
H04N 1/41

(21)Application number : 07-094656

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 20.04.1995

(72)Inventor :
TOYOKURA MAKI
OKAMOTO KIYOSHI
MATSUMOTO YOSHIFUMI

(30)Priority

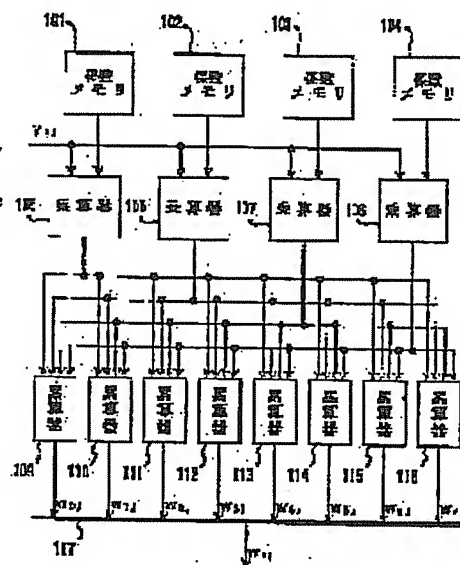
Priority number : 06 96196 Priority date : 10.05.1994 Priority country : JP
06268701 01.11.1994 JP

(54) ORTHOGONAL TRANSFORMATION PROCESSOR

(57)Abstract

PURPOSE: To reduce the circuit size of a one-dimensional IDCT processor.

CONSTITUTION: Eight elements of each row in a matrix consisting of four rows and eight columns whose constitutional elements consist of the absolute values of respective elements of four upper rows in a reverse discrete cosine matrix consisting of eight rows and eight columns for eight-point IDCT processing are respectively stored in four coefficient memories 101 to 104. An input element y_{ij} is supplied to four multipliers 105 to 108 in parallel. Each of the multipliers 105 to 108 executes multiplication between an output from its corresponding coefficient memory out of the memories 101 to 104 and the input element y_{ij} . Eight accumulators 109 to 116 execute parallel accumulation for finding out eight inner products by the use of the results of the multipliers 105 to 108 while decoding the codes of coefficients in an orthogonal transformation matrix. An eight-input selector 117 successively selects the results of the accumulators 109 to 116 and outputs an inner product w_{ij} corresponding to the input element y_{ij} .



LEGAL STATUS

[Date of request for examination] 26.05.1997

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2901896

[Date of registration] 19.03.1999

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-185389

(43)公開日 平成8年(1996)7月16日

(51)Int Cl⁵

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 17/14

G 0 6 T 1/00

G 1 0 L 9/16

G 0 6 F 15/ 332

S

15/ 66

M

審査請求 未請求 請求項の数20 O L (全 19 頁) 最終頁に続く

(21)出願番号 特願平7-94656

(22)出願日 平成7年(1995)4月20日

(31)優先権主張番号 特願平6-96196

(32)優先日 平6(1994)5月10日

(33)優先権主張国 日本 (J P)

(31)優先権主張番号 特願平6-268701

(32)優先日 平6(1994)11月1日

(33)優先権主張国 日本 (J P)

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72)発明者 豊蔵 真木

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72)発明者 岡本 潔

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72)発明者 松本 義史

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

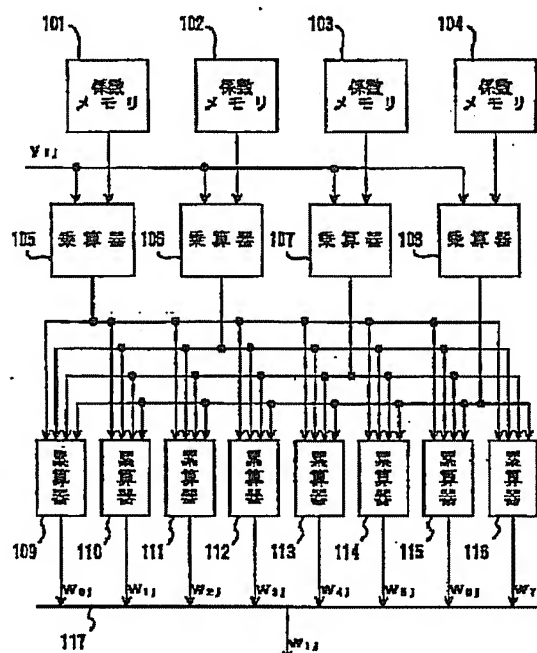
(74)代理人 弁理士 前田 弘 (外2名)

(54)【発明の名称】 直交変換プロセッサ

(57)【要約】

【目的】 1次元IDCTプロセッサの回路規模を低減する。

【構成】 8ポイントIDCT処理のための8行8列の逆離散コサイン行列のうちの上4行の各要素の絶対値を構成要素とした4行8列の行列の各行の8個の要素を4個の係数メモリ101~104にそれぞれ格納する。入力要素 $y_{i,j}$ が4個の乗算器105~108へ並列に供給される。4個の乗算器105~108は、4個の係数メモリ101~104のうちの対応する係数メモリの出力と入力要素 $y_{i,j}$ との乗算をそれぞれ実行する。8個の累算器109~116は、直交変換行列の係数の符号を復元しながら4個の乗算器105~108の結果を用いて8個の内積を求めるための累算を並列に実行する。8入力セクタ117は、8個の累算器109~116の結果を順次選択して、入力要素 $y_{i,j}$ に対応した内積 $w_{i,j}$ を出力する。



【特許請求の範囲】

【請求項1】 2^{n+1} (n は2以上の整数)個の要素からなる入力データに直交変換処理を施すための直交変換プロセッサであって、

直交変換行列を構成する $2^{n+1} \times 2^{n+1}$ 個の係数のうちの $2^n \times 2^{n+1}$ 個の係数の各々の絶対値をそれぞれ 2^{n+1} 個ずつ格納するための第1から第 2^n の係数メモリと、

前記入力データのうちの1個の要素と前記第1から第 2^n の係数メモリの中の対応する係数メモリの中の 2^{n+1} 個の格納係数のうちの1個との乗算をそれぞれ実行するための第1から第 2^n の乗算器と、

前記直交変換行列に対応した 2^{n+1} 個の内積を並列に求めるように、前記直交変換行列の係数の符号を復元しながら前記第1から第 2^n の乗算器の結果を用いた累算をそれぞれ実行するための第1から第 2^{n+1} の累算器と、前記第1から第 2^{n+1} の累算器の結果を前記直交変換プロセッサの出力データの要素として順次選択出力するための 2^{n+1} 入力セクタとを備えたことを特徴とする直交変換プロセッサ。

【請求項2】 請求項1記載の直交変換プロセッサにおいて、 n が2であることを特徴とする直交変換プロセッサ。

【請求項3】 請求項2記載の直交変換プロセッサにおいて、

前記第1から第8の累算器の各々は、前記第1から第4の乗算器のうちの1個の乗算器の結果そのものと、該結果の2の補数とのいずれかを選択出力するための2の補数器と、

前記2の補数器の出力と累算結果との加算を実行するための加算器と、

前記累算結果の初期値として0を用い、かつ前記累算結果の中間値として前記加算器の結果を保持出力するための累算レジスタと、

前記累算レジスタの出力を保持出力するためのバッファレジスタとを備えたことを特徴とする直交変換プロセッサ。

【請求項4】 請求項3記載の直交変換プロセッサにおいて、

前記第1から第8の累算器の各々は、前記第1から第4の乗算器のうちの1個の乗算器の結果を前記2の補数器の入力として選択出力するための4入力セクタを更に備えたことを特徴とする直交変換プロセッサ。

【請求項5】 請求項2記載の直交変換プロセッサにおいて、

前記第1から第8の累算器の各々は、前記第1から第4の乗算器のうちの1個の乗算器の結果そのものと、該結果の1の補数とのいずれかを選択出力するための1の補数器と、

前記1の補数器の出力と累算結果との加算を実行するた

めの加算器と、

前記累算結果の初期値として定数を用い、かつ前記累算結果の中間値として前記加算器の結果を保持出力するための累算レジスタと、

前記累算レジスタの出力を保持出力するためのバッファレジスタとを備えたことを特徴とする直交変換プロセッサ。

【請求項6】 請求項5記載の直交変換プロセッサにおいて、

10 前記第1から第8の累算器の各々は、前記第1から第4の乗算器のうちの1個の乗算器の結果を前記1の補数器の入力として選択出力するための4入力セクタを更に備えたことを特徴とする直交変換プロセッサ。

【請求項7】 2^{n+1} (n は2以上の整数)個の要素からなる入力データに直交変換処理を施すための直交変換プロセッサであって、直交変換行列を構成する $2^{n+1} \times 2^{n+1}$ 個の係数のうちの $(2^n - 1) \times 2^{n+1}$ 個の係数の各々の絶対値をそれぞれ 2^{n+1} 個ずつ格納するための第1から第 $(2^n - 1)$ の係数メモリと、

20 入力として供給された前記入力データの要素を1個ずつ保持出力するための入力レジスタと、

前記入力レジスタの入力と出力とのいずれかを選択出力するための2入力セクタと、

前記2入力セクタの出力と前記第1の係数メモリの中の 2^{n+1} 個の格納係数のうちの1個との乗算を実行するための第1の乗算器と、

前記入力レジスタの出力と前記第2から第 $(2^n - 1)$ の係数メモリの中の対応する係数メモリの中の 2^{n+1} 個の格納係数のうちの1個との乗算をそれぞれ実行するための第2から第 $(2^n - 1)$ の乗算器と、

30 前記第1の乗算器の結果を保持出力するための一時レジスタと、

前記直交変換行列に対応した 2^{n+1} 個の内積を並列に求めるように、前記直交変換行列の係数の符号を復元しながら前記第1から第 $(2^n - 1)$ の乗算器の結果と前記一時レジスタの出力とを用いた累算をそれぞれ実行するための第1から第 2^{n+1} の累算器と、

前記第1から第 2^{n+1} の累算器の結果を前記直交変換プロセッサの出力データの要素として順次選択出力するための 2^{n+1} 入力セクタとを備えたことを特徴とする直交変換プロセッサ。

【請求項8】 請求項7記載の直交変換プロセッサにおいて、

n が2であることを特徴とする直交変換プロセッサ。

【請求項9】 請求項8記載の直交変換プロセッサにおいて、

前記第1から第8の累算器の各々は、前記第1から第3の乗算器の結果と前記一時レジスタの出力とのうちのいずれかを選択出力するための4入力セ

レクタと、

前記4入力セクタの出力そのものと、該出力の2の補数とのいずれかを選択出力するための2の補数器と、

前記2の補数器の出力と累算結果との加算を実行するための加算器と、

前記累算結果の初期値として0を予め保持し、かつ前記累算結果の中間値として前記加算器の結果を保持出力するための累算レジスタと、

前記累算レジスタの出力を保持出力するためのバッファレジスタとを備えたことを特徴とする直交変換プロセッサ。 10

【請求項10】 請求項8記載の直交変換プロセッサにおいて、

前記第1から第8の累算器の各々は、

前記第1から第3の乗算器の結果と前記一時レジスタの出力とのうちのいずれかを選択出力するための4入力セクタと、

前記4入力セクタの出力そのものと、該出力の1の補数とのいずれかを選択出力するための1の補数器と、

前記1の補数器の出力と累算結果との加算を実行するための加算器と、 20

前記累算結果の定数初期値を予め保持し、かつ前記累算結果の中間値として前記加算器の結果を保持出力するための累算レジスタと、

前記累算レジスタの出力を保持出力するためのバッファレジスタとを備えたことを特徴とする直交変換プロセッサ。

【請求項11】 2^{n+1} (n は2以上の整数)個の要素からなる入力データに直交変換処理を施すための直交変換プロセッサであって、

前記入力データの連続する 2^{n+1} 個の要素を一括して保持出力するための入力バッファと、

前記入力バッファから前記 2^{n+1} 個の要素のうちの第1番目の要素と第 $(2^n + 1)$ 番目の要素とを入力して2個の定数乗算結果を並列出力するための定数乗算回路と、

前記入力バッファから他の $(2^{n+1} - 2)$ 個の要素を入力して、直交変換行列に対応した 2^{n+1} 個の部分内積を順次出力するための分布演算回路と、

前記直交変換プロセッサの出力データの要素を求めるように、前記定数乗算回路の2個の出力と前記分布演算回路の出力との合成演算を実行するための合成演算回路とを備えたことを特徴とする直交変換プロセッサ。 40

【請求項12】 請求項11記載の直交変換プロセッサにおいて、

n が2であることを特徴とする直交変換プロセッサ。

【請求項13】 請求項12記載の直交変換プロセッサにおいて、

前記入力バッファは、前記入力データの連続する8個の要素の各々を保持出力するための8個のレジスタを備え 50

たことを特徴とする直交変換プロセッサ。

【請求項14】 請求項12記載の直交変換プロセッサにおいて、

前記定数乗算回路は、

前記入力データの連続する8個の要素のうちの第1番目の要素を保持出力するための第1の入力レジスタと、

前記入力データの連続する8個の要素のうちの第5番目の要素を保持出力するための第2の入力レジスタと、

前記第1の入力レジスタの出力と前記第2の入力レジスタの出力とを順次選択出力するための2入力セクタと、

前記2入力セクタの出力を用いて、前記第1の入力レジスタの出力の第1の定数乗算と、前記第2の入力レジスタの第2の定数乗算とを順次実行するための乗算器と、

前記第1の定数乗算の結果を保持出力するための第1の一時レジスタと、

前記第2の定数乗算の結果を保持出力するための第2の一時レジスタと、

前記第1の一時レジスタの出力を保持出力するための第1のバッファレジスタと、

前記第2の一時レジスタの出力を保持出力するための第2のバッファレジスタとを備え、かつ前記合成演算回路は、前記第1のバッファレジスタの出力と前記分布演算回路の出力とを各々加算入力とし、かつ前記第2のバッファレジスタの出力を加減算入力とした加減算を実行するための3入力加減算器を備えたことを特徴とする直交変換プロセッサ。

【請求項15】 請求項12記載の直交変換プロセッサにおいて、 30

前記定数乗算回路は、

前記入力データの連続する8個の要素のうちの第1番目の要素を保持出力するための第1の入力レジスタと、

前記入力データの連続する8個の要素のうちの第5番目の要素を保持出力するための第2の入力レジスタと、

前記第1の入力レジスタの出力と前記第2の入力レジスタの出力との加算及び減算を順次実行するための2入力加減算器と、

前記2入力加減算器の加算結果の第1の定数乗算と、前記2入力加減算器の減算結果の第2の定数乗算とを順次実行するための乗算器と、

前記第1の定数乗算の結果を保持出力するための第1の一時レジスタと、

前記第2の定数乗算の結果を保持出力するための第2の一時レジスタと、

前記第1の一時レジスタの出力を保持出力するための第1のバッファレジスタと、

前記第2の一時レジスタの出力を保持出力するための第2のバッファレジスタとを備え、かつ前記合成演算回路は、 50

前記第1及び第2のバッファレジスタの出力のいずれかを選択出力するための2入力セクタと、

前記2入力セクタの出力と前記分布演算回路の出力との加算を実行するための2入力加算器とを備えたことを特徴とする直交変換プロセッサ。

【請求項16】 請求項12記載の直交変換プロセッサにおいて、

前記分布演算回路は、

前記入力データの連続する8個の要素のうちの第2、第3、第4、第6、第7及び第8番目の要素を保持し、かつ該6要素の各々の最下位ビットを集めて第1のビットスライスワードとし、該6要素の各々の最下位ビットより1桁上位のビットを集めて第2のビットスライスワードとするように、該6要素の各々の最下位2ビットを次々とシフトアウトするための6個のシフトレジスタと、前記直交変換行列に対応した8個の部分内積を並列に求めるように、前記第1及び第2のビットスライスワードに基づく積和演算をそれぞれ実行するための8個の6ビット入力RACと、

前記8個の6ビット入力RACの結果を保持出力するための8個のバッファレジスタと、

前記8個のバッファレジスタの出力を順次選択出力するための8入力セクタとを備えたことを特徴とする直交変換プロセッサ。

【請求項17】 請求項16記載の直交変換プロセッサにおいて、

前記8個の6ビット入力RACの各々は、

前記第1のビットスライスワードをアドレスとして索引されるように、前記直交変換行列に基づくベクトル内積の部分積を格納するための第1のROMと、

前記第2のビットスライスワードをアドレスとして索引されるように、前記直交変換行列に基づくベクトル内積の部分積を格納するための第2のROMと、

前記第1のROMから索引された部分積を第1の加算入力とし、前記第2のROMから索引された部分積を加減算入力とし、かつ累算結果を第2の加算入力とした加減算を実行するための3入力加減算器と、

前記3入力加減算器の結果を左シフト出力するためのシフタと、

前記累算結果の初期値として0を予め保持し、かつ前記累算結果の中間値として前記シフタの出力を保持出力するための累算レジスタとを備えたことを特徴とする直交変換プロセッサ。

【請求項18】 請求項12記載の直交変換プロセッサにおいて、

前記分布演算回路は、

前記入力データの連続する8個の要素のうちの第2、第4、第6及び第8番目の要素を保持し、かつ該4要素の各々の最下位ビットを集めて第1のビットスライスワードとし、該4要素の各々の最下位ビットより1桁上位の

ビットを集めて第2のビットスライスワードとするように、該4要素の各々の最下位2ビットを次々とシフトアウトするための4個のシフトレジスタと、

前記入力データの連続する8個の要素のうちの第3及び第7番目の要素を保持し、かつ該2要素の各々の最下位ビットを集めて第3のビットスライスワードとし、該2要素の各々の最下位ビットより1桁上位のビットを集めて第4のビットスライスワードとするように、該2要素の各々の最下位2ビットを次々とシフトアウトするための2個のシフトレジスタと、

前記直交変換行列に対応した4個の部分内積を並列に求めるように、前記第1及び第2のビットスライスワードに基づく積和演算をそれぞれ実行するための4個の4ビット入力RACと、

前記直交変換行列に対応した4個の部分内積を並列に求めるように、前記第3及び第4のビットスライスワードに基づく積和演算をそれぞれ実行するための4個の2ビット入力RACと、

前記4個の4ビット入力RACの結果を保持出力するための第1から第4のバッファレジスタと、

前記4個の2ビット入力RACの結果を保持出力するための第5から第8のバッファレジスタと、

前記第1から第4のバッファレジスタの出力を順次選択出力するための第1の4入力セクタと、

前記第5から第8のバッファレジスタの出力を順次選択出力するための第2の4入力セクタと、

前記第1の4入力セクタの出力を加減算入力とし、かつ前記第2の4入力セクタの出力を加算入力とした加減算を実行するための2入力加減算器とを備えたことを特徴とする直交変換プロセッサ。

【請求項19】 請求項18記載の直交変換プロセッサにおいて、

前記4個の4ビット入力RACの各々は、

前記第1のビットスライスワードをアドレスとして索引されるように、前記直交変換行列に基づくベクトル内積の部分積を格納するための第1のROMと、

前記第2のビットスライスワードをアドレスとして索引されるように、前記直交変換行列に基づくベクトル内積の部分積を格納するための第2のROMと、

前記第1のROMから索引された部分積を第1の加算入力とし、前記第2のROMから索引された部分積を加減算入力とし、かつ累算結果を第2の加算入力とした加減算を実行するための3入力加減算器と、

前記3入力加減算器の結果を左シフト出力するためのシフタと、

前記累算結果の初期値として0を予め保持し、かつ前記累算結果の中間値として前記シフタの出力を保持出力するための累算レジスタとを備えたことを特徴とする直交変換プロセッサ。

【請求項20】 請求項18記載の直交変換プロセッサ

において、

前記4個の2ビット入力RACの各々は、
前記第3のビットスライズワードをアドレスとして索引されるように、前記直交変換行列に基づくベクトル内積の部分積を格納するための第1のROMと、
前記第4のビットスライズワードをアドレスとして索引されるように、前記直交変換行列に基づくベクトル内積の部分積を格納するための第2のROMと、
前記第1のROMから索引された部分積を第1の加算入力とし、前記第2のROMから索引された部分積を加減算入力とし、かつ累算結果を第2の加算入力とした加減算を実行するための3入力加減算器と、
前記3入力加減算器の結果を左シフト出力するためのシフトと、
前記累算結果の初期値として0を予め保持し、かつ前記累算結果の中間値として前記シフトの出力を保持出力するための累算レジスタとを備えたことを特徴とする直交変換プロセッサ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、画像情報処理に好適に利用される直交変換プロセッサに関するものである。

【0002】

【従来の技術】近年、2次元画像データの高性能圧縮符号化方式の重要な一部分として、直交変換を実現する小規模な回路が要求されている。符号器では、順方向の直交変換、例えば離散コサイン変換 (discrete cosine tr*

$$d_{10} = 1 / (2 \cdot 2^{0.5}), i = 0 \sim 7$$

$$d_{11} = (1/2) \cos \{ (2i+1) j \pi / 16 \}, i = 0 \sim 7, j = 1 \sim 7$$

... (1)

である。行列Yの2次元IDCTは DYD^T である。ここに、 D^T は行列Dの転置行列である。行列Yの1次元IDCTすなわち行列積DYを計算するための1次元IDCTプロセッサと、転置手段とを用いれば、中間行列 $X = (DY)^T$ が容易に求められる。最終結果 DYD^T も同様にして求められる。なぜなら、 $DYD^T = (D(DY)^T)^T = (DX)^T$ であるからである。つま

$$w_{11} = \sum_{k=0}^7 d_{1k} y_{k1}, i = 0 \sim 7, j = 0 \sim 7 \quad \dots (2)$$

である。要素 w_{11} は、行列Dの第1行と行列Yの第1列との内積であって、8個の積の和である。この要素 w_{11} を求める処理は、8ポイントIDCT処理と呼ばれる。

【0006】8個の乗算器と8個の累算器とを備えた1★

$$w_{01} = \sum_{k=0}^7 d_{0k} y_{k1}$$

$$w_{11} = \sum_{k=0}^7 d_{1k} y_{k1}$$

$$w_{21} = \sum_{k=0}^7 d_{2k} y_{k1}$$

$$w_{31} = \sum_{k=0}^7 d_{3k} y_{k1}$$

$$w_{41} = \sum_{k=0}^7 d_{4k} y_{k1}$$

$$w_{51} = \sum_{k=0}^7 d_{5k} y_{k1}$$

$$w_{61} = \sum_{k=0}^7 d_{6k} y_{k1}$$

$$w_{71} = \sum_{k=0}^7 d_{7k} y_{k1}$$

... (3)

* ansform : 略してDCT)、離散サイン変換 (discrete sine transform : 略してDST) などが利用される。復号器では、逆方向の直交変換、例えば逆離散コサイン変換 (inverse discrete cosine transform : 略してIDCT)、逆離散サイン変換 (inverse discrete sine transform : 略してIDST) などが利用される。

【0003】米国特許4,791,598には、2個の1次元DCTプロセッサと、その間に介在した転置メモリとで構成された2次元DCTプロセッサが開示されている。2個の1次元DCTプロセッサの各々は、乗算器を用いずにROM (read only memory) を用いてベクトル内積を求めるための分布演算 (distributed arithmetic : 略してDA) 回路を内蔵したものである。DA回路は、複数個のROM/累算器 (ROM and accumulator : 略してRAC) を備えている。各RACは、離散コサイン行列に基づくベクトル内積の部分積をルックアップテーブルの形式で格納したROMと、該ROMからビットスライズワードをアドレスとして順次索引された部分積を桁合わせ加算して入力ベクトルに対応するベクトル内積を得るための累算器とを有するものである。このような2次元DCTプロセッサの構成は、2次元IDCTプロセッサに転用可能である。

【0004】 8×8 要素からなる入力データに2次元IDCT処理を施すものとする。入力データは、要素 y_{ij} ($i = 0 \sim 7, j = 0 \sim 7$) を持つ8行8列の行列Yで表わされる。また、8行8列の逆離散コサイン行列Dを考える。行列Dの各要素 d_{ij} は、

※り、行列積DYを計算するための1次元IDCTプロセッサは、2次元IDCTを実現する上で重要な役割を担っている。

【0005】行列Yの第j列に関する1次元IDCTの結果は、8行8列の行列Wの第j列で表わされる。ここに、行列Wの各要素 w_{11} は、

★次元IDCTプロセッサによれば、行列Wの第j列を構成する8個の内積 $w_{01}, w_{11}, w_{21}, w_{31}, w_{41}, w_{51}, w_{61}, w_{71}$ を並列に計算することができる。ここに、

である。

【0007】

【発明が解決しようとする課題】上記8個の乗算器を備えた1次元IDCTプロセッサは、VLSI (very large scale integration) への実装に際して乗算器がチップ上の大きな面積を占める問題があった。

【0008】また、式(3)で表わされた8個の内積の並列計算を上記従来のDA回路で実現する場合には、大きいROMサイズを要するという問題があった。

【0009】本発明の目的は、1次元IDCTプロセッサなどの直交変換プロセッサの回路規模を低減することにある。

【0010】

【課題を解決するための手段】上記の目的を達成するため、本発明に係る第1の直交変換プロセッサは、逆離散コサイン行列や逆離散サイン行列の要素の規則性に鑑みて乗算器の個数を低減し、各乗算器の結果を複数の累算器へ分配することとしたものである。

【0011】また、本発明に係る第2の直交変換プロセッサは、複数の内積計算の各々を2個の定数乗算と1個*20

$$t_0 = 1 / (2 \cdot 2^{0.5})$$

$$t_n = (1/2) \cos(n\pi/16), n=1 \sim 7 \quad \dots (4)$$

のように定義する。すると、前記の式(3)で表わされた8個の内積の計算は、コサイン関数の対称性を利用して、図1のように表現される。

【0016】図1の行列演算では、符号(±)を度外視すると、 y_{0i} に掛けるべき係数は t_0 であり、 y_{1i} に掛けるべき係数は t_1, t_3, t_5, t_7 であり、 y_{2i} に掛けるべき係数は t_2, t_6 であり、 y_{3i} に掛けるべき係数は t_4, t_7, t_1, t_5 であり、 y_{4i} に掛けるべき係数は t_0 であり、 y_{5i} に掛けるべき係数は t_1, t_3, t_5, t_7 であり、 y_{6i} に掛けるべき係数は t_2, t_6 であり、 y_{7i} に掛けるべき係数は t_4, t_7, t_1, t_5 である。したがって、図2に示すように、入力データの8個の要素 y_{ji} ($i=0 \sim 7$) が順次供給されるとき、1サイクルに最大4個の乗算を実行すればよい。図2の手順で用いられる係数行列Eを図3に示す。係数行列Eは、図1中の8行8列の逆離散コサイン行列のうちの第0行から第3行までの各要素の絶対値を構成要素とした4行8列の行列である。

【0017】本発明の第1の実施例に係る1次元IDCTプロセッサの構成を図4に示す。この構成は、図3の係数行列Eを採用したものである。図4において、101~104は第1~第4の係数メモリ、105~108は第1~第4の乗算器、109~116は第1~第8の累算器、117は8入力セクタである。第1の係数メモリ101には行列Eの第0行の8個の要素が、第2の係数メモリ102には行列Eの第1行の8個の要素が、第3の係数メモリ103には行列Eの第2行の8個の要素が、第4の係数メモリ104には行列Eの第3行の8

*の部分内積の計算とに分割し、2個の定数乗算を定数乗算回路で実行することとしたものである。しかも、複数の部分内積の計算をDA回路で並列実行することとした。

【0012】

【作用】本発明に係る第1の直交変換プロセッサによれば、例えば8ポイントIDCT処理の場合に従来は8個の乗算器を要したところ、乗算器数が4又は3に低減される。

【0013】また、本発明に係る第2の直交変換プロセッサによれば、定数乗算回路中の2個又は1個の乗算器を要するのみである。また、内積計算の一部が定数乗算回路で実行されるので、DA回路のROMサイズが低減される。

【0014】

【実施例】以下、図面を参照しながら、本発明の実施例に係る1次元IDCTプロセッサについて説明する。

【0015】(実施例1) まず、 t_n ($n=0 \sim 7$) を、

個の要素がそれぞれ格納されている。入力端子から、2の補数表示の2進数データ y_{ji} ($i=0 \sim 7, j=0 \sim 7$) が、 $y_{00} \sim y_{70}, y_{01} \sim y_{71}, \dots, y_{07} \sim y_{77}$ の順序で第1~第4の乗算器105~108へ供給される。第1の乗算器105は y_{1i} と第1の係数メモリ101の出力との乗算を、第2の乗算器106は y_{1i} と第2の係数メモリ102の出力との乗算を、第3の乗算器107は y_{1i} と第3の係数メモリ103の出力との乗算を、第4の乗算器108は y_{1i} と第4の係数メモリ104の出力との乗算をそれぞれ実行するものである。第1~第8の累算器109~116は、第1~第4の乗算器105~108の結果を用いて8個の内積 $w_{0i}, w_{1i}, w_{2i}, w_{3i}, w_{4i}, w_{5i}, w_{6i}, w_{7i}$ を求めるための累算を並列に実行するものである。8入力セクタ117は、第1~第8の累算器109~116の結果を順次選択して、データ w_{ji} ($i=0 \sim 7, j=0 \sim 7$) を、 $w_{00} \sim w_{70}, w_{01} \sim w_{71}, \dots, w_{07} \sim w_{77}$ の順序で出力するものである。

【0018】図4中の w_{1i} を求めるための累算器110の内部構成を図5に示す。図5において、201は4入力セクタ、202は2の補数器、203は加算器、204は累算レジスタ、205はバッファレジスタである。4入力セクタ201は、第1~第4の乗算器105~108の結果のうちの1つを選択するものである。2の補数器202は、 i の値に応じて、4入力セクタ201の出力をそのまま通過させたり、4入力セクタ201の出力の2の補数を出力したりするものである。具体的には、図1中の逆離散コサイン行列のうちの第1

行($t_0, t_1, t_2, \dots, -t_7, -t_4, -t_1, -t_2, -t_5$)と入力データベクトル($y_{01}, y_{11},$

$y_{21}, y_{31}, y_{41}, y_{51}, y_{61}, y_{71}$)との内積 w_{11} の計算に対応して、 $i=0, 1, 2$ のサイクルでは4入力セクタ201の出力をそのまま通過させ、 $i=3,$

4, 5, 6, 7のサイクルでは4入力セクタ201の出力の2の補数を出力するように制御される。データ x の2の補数は、 x の全てのビットを反転させたうえ、1を加算することによって求められる。加算器203は、2の補数器202の結果と累算レジスタ204の保持出力との和を求めるものである。累算レジスタ204の保持内容は予め0に初期化されたうえ、加算器203の結果に書き換えられる。バッファレジスタ205は、当該1次元IDCTプロセッサのパイプライン動作を保証するように、累算レジスタ204の出力を保持するものである。図4中の他の累算器の内部構成も図5と同様である。

【0019】以下、図4及び図5に基づいて、本発明の第1の実施例に係る1次元IDCTプロセッサの動作を説明する。

【0020】第1のサイクルでは、入力端子からデータ y_{00} が供給される。一方、係数メモリ101~104からそれぞれ t_0, t_1, t_2, t_3 が読み出され、乗算器105~108により4個の積 $t_0 y_{00}, t_1 y_{00}, t_2 y_{00}, t_3 y_{00}$ が並列に計算される。次に、累算器109~116の4入力セクタ201により、4個の乗算器105~108の結果のうちの1個がそれぞれ選択される。この場合、4個の乗算器105~108の結果は全て同じであるので、どれを選択してもよい。累算器109~116の2の補数器202は、それぞれ4入力セクタ201の出力をそのまま通過させる。累算器109~116の加算器203は、2の補数器202の結果と、予め0に初期化された累算レジスタ204の出力との和を計算し、その加算結果を累算レジスタ204にそれぞれ書き込む。この結果、累算器109~116の全ての累算レジスタ204に同じ積 $t_0 y_{00}$ が格納される。

【0021】第2のサイクルでは、入力端子からデータ y_{10} が供給される。一方、係数メモリ101~104からそれぞれ t_1, t_2, t_3, t_4 が読み出され、乗算器105~108により4個の積 $t_1 y_{10}, t_2 y_{10}, t_3 y_{10}, t_4 y_{10}$ が並列に計算される。次に、累算器109~116の4入力セクタ201により、4個の乗算器105~108の結果のうちの1個がそれぞれ選択される。この場合、第1の累算器109では第1の乗算器105の結果 $t_1 y_{10}$ が、第2の累算器110では第2の乗算器106の結果 $t_2 y_{10}$ が、第3の累算器111では第3の乗算器107の結果 $t_3 y_{10}$ が、第4の累算器112では第4の乗算器108の結果 $t_4 y_{10}$ が、第5の累算器113では第4の乗算器108の結

果 $t_7 y_{10}$ が、第6の累算器114では第3の乗算器107の結果 $t_1 y_{10}$ が、第7の累算器115では第2の乗算器106の結果 $t_2 y_{10}$ が、第8の累算器116では第1の乗算器105の結果 $t_1 y_{10}$ がそれぞれ選択される。第1~第4の累算器109~112の2の補数器202は、それぞれ4入力セクタ201の出力をそのまま通過させる。第5~第8の累算器113~116の2の補数器202は、それぞれ4入力セクタ201の出力の2の補数を出力する。累算器109~116の加算器203は、2の補数器202の結果と累算レジスタ204の出力との和を計算し、その加算結果を累算レジスタ204にそれぞれ書き込む。この結果、第1の累算器109では $t_0 y_{00} + t_1 y_{10}$ が、第2の累算器110では $t_1 y_{00} + t_2 y_{10}$ が、第3の累算器111では $t_2 y_{00} + t_3 y_{10}$ が、第4の累算器112では $t_3 y_{00} + t_4 y_{10}$ が、第5の累算器113では $t_4 y_{00} - t_7 y_{10}$ が、第6の累算器114では $t_1 y_{00} - t_2 y_{10}$ が、第7の累算器115では $t_2 y_{00} - t_3 y_{10}$ が、第8の累算器116では $t_3 y_{00} - t_4 y_{10}$ がそれぞれ累算レジスタ204に格納される。

【0022】第3から第8のサイクルでは、入力端子からデータ $y_{20}, y_{30}, y_{40}, y_{50}, y_{60}, y_{70}$ が順次供給される。したがって、第8サイクルの終りには、累算器109~116の累算レジスタ204に、8個の内積 $w_{00}, w_{10}, w_{20}, w_{30}, w_{40}, w_{50}, w_{60}, w_{70}$ が格納される。

【0023】第9のサイクルでは、入力端子からデータ y_{01} が供給されて上記第1のサイクルと同様の処理が実行されるとともに、累算器109~116の累算レジスタ204の保持内容 $w_{00}, w_{10}, w_{20}, w_{30}, w_{40}, w_{50}, w_{60}, w_{70}$ がバッファレジスタ205へそれぞれ転送される。そして、8入力セクタ117は、第1の累算器109の出力 w_{00} を選択出力する。

【0024】第10のサイクルでは、入力端子からデータ y_{11} が供給されて、上記第2のサイクルと同様の処理が実行される。8入力セクタ117は、第2の累算器110の出力 w_{10} を選択出力する。

【0025】以下同様の処理を繰り返すことにより、連続的に供給される入力データ $y_{00} \sim y_{70}, y_{01} \sim y_{71}, \dots, y_{07} \sim y_{77}$ に対応した出力データ $w_{00} \sim w_{70}, w_{01} \sim w_{71}, \dots, w_{07} \sim w_{77}$ が連続して得られる。

【0026】図5の累算器110の変形例を図6に示す。図6の例では、上記2の補数器202に代えて1の補数器212が用いられる。1の補数器212は、上記 i の値に応じて、4入力セクタ201の出力をそのまま通過させたり、4入力セクタ201の出力の1の補数を出力したりするものである。具体的には、図1中の逆離散コサイン行列のうちの第1行($t_0, t_1, t_2, \dots, -t_7, -t_4, -t_1, -t_2, -t_5$)と入力データベクトル($y_{01}, y_{11}, y_{21}, y_{31}, y_{41}, y_{51}, y_{61}, y_{71}$)が、第6の累算器114では第3の乗算器107の結果 $t_1 y_{10}$ が、第7の累算器115では第2の乗算器106の結果 $t_2 y_{10}$ が、第8の累算器116では第1の乗算器105の結果 $t_1 y_{10}$ がそれぞれ選択される。第1~第4の累算器109~112の2の補数器202は、それぞれ4入力セクタ201の出力をそのまま通過させる。第5~第8の累算器113~116の2の補数器202は、それぞれ4入力セクタ201の出力の2の補数を出力する。累算器109~116の加算器203は、2の補数器202の結果と累算レジスタ204の出力との和を計算し、その加算結果を累算レジスタ204にそれぞれ書き込む。この結果、第1の累算器109では $t_0 y_{00} + t_1 y_{10}$ が、第2の累算器110では $t_1 y_{00} + t_2 y_{10}$ が、第3の累算器111では $t_2 y_{00} + t_3 y_{10}$ が、第4の累算器112では $t_3 y_{00} + t_4 y_{10}$ が、第5の累算器113では $t_4 y_{00} - t_7 y_{10}$ が、第6の累算器114では $t_1 y_{00} - t_2 y_{10}$ が、第7の累算器115では $t_2 y_{00} - t_3 y_{10}$ が、第8の累算器116では $t_3 y_{00} - t_4 y_{10}$ がそれぞれ累算レジスタ204に格納される。

10

20

30

40

50

y_{61}, y_{71}) との内積 w_{11} の計算に対応して、 $i = 0, 1, 2$ のサイクルでは4入力セクタ201の出力をそのまま通過させ、 $i = 3, 4, 5, 6, 7$ のサイクルでは4入力セクタ201の出力の1の補数を出力するように制御される。データ x の1の補数は、 x の全てのビットを反転させることによって求められる。累算レジスタ204の初期値は、上記逆離散コサイン行列の第1行を構成する8個の要素のうちの負の要素の数、すなわち5に設定される。

【0027】以上のとおり、第1の実施例によれば、乗算器の数が4に低減された1次元IDCTプロセッサを実現できる。なお、累算器109~116の4入力セクタ201(図5及び図6)を省略して固定配線を採用してもよい。この場合、第1及び第8の累算器109、116へは第1の乗算器105の結果のみが、第2及び第7の累算器110、115へは第2の乗算器106の結果のみが、第3及び第6の累算器111、114へは第3の乗算器107の結果のみが、第4及び第5の累算器112、113へは第4の乗算器108の結果のみがそれぞれ供給される。

【0028】(実施例2) 図1の行列演算を遂行するためには、図2に示すように、8サイクルで22個の乗算を実行する必要がある。1サイクルの平均乗算個数は2.75である。そこで、第2の実施例では、図7に示すように、入力データの8個の要素 y_{i1} ($i = 0 \sim 7$) が順次供給されるとき、1サイクルに最大3個の乗算を実行することとした。そのため、入力要素を保持するためのレジスタを設け、あるサイクルの入力要素に加えて前サイクルの入力要素を利用できるようにした。すなわち、あるサイクルでは第1群の係数 t_1, t_2 と要素 y_{01}, y_{01} との乗算が、次のサイクルでは第2群の係数 t_3, t_4 と要素 y_{11}, y_{11} との乗算が、次のサイクルでは第3群の係数 t_5, t_6 と要素 y_{21}, y_{21} との乗算が、次のサイクルでは第4群の係数 t_7, t_8 と要素 y_{31}, y_{31} との乗算が、次のサイクルでは第5群の係数 t_9, t_{10} と要素 y_{41}, y_{41} との乗算が、次のサイクルでは第6群の係数 t_{11}, t_{12} と要素 y_{51}, y_{51} との乗算が、次のサイクルでは第7群の係数 t_{13}, t_{14} と要素 y_{61}, y_{61} との乗算が、次のサイクルでは第8群の係数 t_{15}, t_{16} と要素 y_{71}, y_{71} との乗算がそれぞれ実行される。図7の手順で用いられる係数行列 G を図8に示す。係数行列 G は、上記第1群の係数を含む3個の係数 t_1, t_2, t_3 を持つ第0列と、上記第2群の係数 t_4, t_5, t_6 を持つ第1列と、上記第3群の係数 t_7, t_8, t_9 を持つ第2列と、上記第4群の係数 t_{10}, t_{11}, t_{12} を持つ第3列と、上記第5群の係数を含む3個の係数 t_{13}, t_{14}, t_{15} を持つ第4列と、上記第6群の係数 t_{16}, t_{17}, t_{18} を持つ第5列と、上記第7群の係数 t_{19}, t_{20}, t_{21} を持つ第6列と、上記第8群の係数 t_{22}, t_{23}, t_{24} を持つ第7列とで構成された3

行8列の行列である。

【0029】本発明の第2の実施例に係る1次元IDCTプロセッサの構成を図9に示す。この構成は、図8の係数行列 G を採用したものである。図9において、301は入力レジスタ、302~304は第1~第3の係数メモリ、305は2入力セクタ、306~308は第1~第3の乗算器、309は一時レジスタ、310~317は第1~第8の累算器、318は8入力セクタである。第1の係数メモリ302には行列 G の第0行の8個の要素が、第2の係数メモリ303には行列 G の第1行の8個の要素が、第3の係数メモリ304には行列 G の第2行の8個の要素がそれぞれ格納されている。入力端子から、2の補数表示の2進数データ y_{i1} ($i = 0 \sim 7, j = 0 \sim 7$) が、 $y_{00} \sim y_{70}, y_{01} \sim y_{71}, \dots, y_{07} \sim y_{77}$ の順序で、入力レジスタ301及び2入力セクタ305へ供給される。2入力セクタ305は、入力端子から直接供給されたデータと入力レジスタ301の出力データとのいずれかを選択するものである。第1の乗算器306は2入力セクタ305の出力と第1の係数メモリ302の出力との乗算を、第2の乗算器307は入力レジスタ301の出力と第2の係数メモリ303の出力との乗算を、第3の乗算器308は入力レジスタ301の出力と第2の係数メモリ304の出力との乗算をそれぞれ実行するものである。一時レジスタ309は、第1の乗算器306の出力を一時保持するものである。第1~第8の累算器310~317は、一時レジスタ309の出力データと第1~第3の乗算器306~308の結果とを用いて8個の内積 $w_{01}, w_{11}, w_{21}, w_{31}, w_{41}, w_{51}, w_{61}, w_{71}$ を求めるための累算を並列に実行するものであって、各々の内部構成は図5又は図6のとおりである。8入力セクタ318は、第1~第8の累算器310~317の結果を順次選択して、データ w_{i1} ($i = 0 \sim 7, j = 0 \sim 7$) を、 $w_{00} \sim w_{70}, w_{01} \sim w_{71}, \dots, w_{07} \sim w_{77}$ の順序で出力するものである。

【0030】以下、図9及び図5に基づいて、本発明の第2の実施例に係る1次元IDCTプロセッサの動作を説明する。

【0031】第1のサイクルでは、入力端子からデータ y_{00} が供給される。更に、第1のサイクルの終わりで、該データ y_{00} が入力レジスタ301に書き込まれる。

【0032】第2のサイクルでは、入力端子からデータ y_{10} が供給され、該データ y_{10} が2入力セクタ305により選択される。一方、係数メモリ302~304からそれぞれ t_1, t_2, t_3 が読み出され、乗算器306~308により3個の積 $t_1 y_{00}, t_2 y_{00}, t_3 y_{00}$ が並列に計算される。次に、累算器310~317の4入力セクタ201により、第2及び第3の乗算器307、308の結果のうちの1個がそれぞれ選択される。この場合、第2及び第3の乗算器307、308の

結果は同じであるので、いずれを選択してもよい。累算器310~317の2の補数器202は、それぞれ4入力セクタ201の出力をそのまま通過させる。累算器310~317の加算器203は、2の補数器202の結果と、予め0に初期化された累算レジスタ204の出力との和を計算し、その加算結果を累算レジスタ204にそれぞれ書き込む。この結果、累算器310~317の全ての累算レジスタ204に、同じ積 t, y_{10} が格納される。更に、第2のサイクルの終わりで、データ y_{10} が入力レジスタ301に書き込まれ、かつ第1の乗算器306の結果 t, y_{10} が一時レジスタ309に書き込まれる。

【0033】第3のサイクルでは、入力端子からデータ y_{10} が供給される。2入力セクタ305は、入力レジスタ301の出力データ y_{10} を選択する。一方、係数メモリ302~304からそれぞれ t_1, t_2, t_3 が読み出され、乗算器306~308により3個の積 $t, y_{10}, t_1 y_{10}, t_2 y_{10}, t_3 y_{10}$ が並列に計算される。次に、累算器310~317の4入力セクタ201により、一時レジスタ309の出力データと3個の乗算器306~308の結果とのうちの1個がそれぞれ選択される。この場合、第1の累算器310では一時レジスタ309の出力データ t, y_{10} が、第2の累算器311では第1の乗算器306の結果 t, y_{10} が、第3の累算器312では第2の乗算器307の結果 t, y_{10} が、第4の累算器313では第3の乗算器308の結果 t, y_{10} が、第5の累算器314では第3の乗算器308の結果 t, y_{10} が、第6の累算器315では第2の乗算器307の結果 t, y_{10} が、第7の累算器316では第1の乗算器306の結果 t, y_{10} が、第8の累算器317では一時レジスタ309の出力データ t, y_{10} がそれぞれ選択される。第1~第4の累算器310~313の2の補数器202は、それぞれ4入力セクタ201の出力をそのまま通過させる。第5~第8の累算器314~317の2の補数器202は、それぞれ4入力セクタ201の出力の2の補数を出力する。累算器310~317の加算器203は、2の補数器202の結果と累算レジスタ204の出力との和を計算し、その加算結果を累算レジスタ204にそれぞれ書き込む。この結果、第1の累算器310では $t, y_{10} + t_1 y_{10}$ が、第2の累算器311では $t, y_{10} + t_2 y_{10}$ が、第3の累算器312では $t, y_{10} + t_3 y_{10}$ が、第4の累算器313では $t, y_{10} + t_3 y_{10} - t, y_{10}$ が、第5の累算器314では $t, y_{10} - t, y_{10} - t_1 y_{10}$ が、第6の累算器315では $t, y_{10} - t, y_{10} - t_2 y_{10}$ が、第7の累算器316では $t, y_{10} - t, y_{10} - t_3 y_{10}$ が、第8の累算器317では $t, y_{10} - t, y_{10} + t, y_{10}$ がそれぞれ累算レジスタ204に格納される。更に、第4のサイクルの終わりで、データ y_{10} が入力レジスタ301に書き込まれ、かつ第1の乗算器306の結果 t, y_{10} が一時レジスタ309に書き込まれる。

【0034】第4のサイクルでは、入力端子からデータ y_{10} が供給され、該データ y_{10} が2入力セクタ305により選択される。一方、係数メモリ302~304からそれぞれ t_1, t_2, t_3 が読み出され、乗算器306~308により3個の積 $t, y_{10}, t_1 y_{10}, t_2 y_{10}, t_3 y_{10}$ が並列に計算される。次に、累算器310~317の4入力セクタ201により、第2及び第3の乗算器307, 308の結果のうちの1個がそれぞれ選択される。この場合、第1、第4、第5及び第8の累算器310, 313, 314, 317では第2の乗算器307の結果 t, y_{10} が、第2、第3、第6及び第7の累算器311, 312, 315, 316では第3の乗算器308の結果 t, y_{10} がそれぞれ選択される。第1、第2、第7及び第8の累算器310, 311, 316, 317の2の補数器202は、それぞれ4入力セクタ201の出力をそのまま通過させる。第3~第6の累算器312~315の2の補数器202は、それぞれ4入力セクタ201の出力の2の補数を出力する。累算器310~317の加算器203は、2の補数器202の結果と累算レジスタ204の出力との和を計算し、その加算結果を累算レジスタ204にそれぞれ書き込む。この結果、第1の累算器310では $t, y_{10} + t_1 y_{10} + t_2 y_{10}$ が、第2の累算器311では $t, y_{10} + t, y_{10} + t, y_{10}$ が、第3の累算器312では $t, y_{10} + t, y_{10} - t, y_{10}$ が、第4の累算器313では $t, y_{10} + t, y_{10} - t, y_{10}$ が、第5の累算器314では $t, y_{10} - t, y_{10} - t, y_{10}$ が、第6の累算器315では $t, y_{10} - t, y_{10} - t, y_{10}$ が、第7の累算器316では $t, y_{10} - t, y_{10} + t, y_{10}$ が、第8の累算器317では $t, y_{10} - t, y_{10} + t, y_{10}$ がそれぞれ累算レジスタ204に格納される。更に、第4のサイクルの終わりで、データ y_{10} が入力レジスタ301に書き込まれ、第1の乗算器306の結果 t, y_{10} が一時レジスタ309に書き込まれる。

【0035】第5のサイクルでは、入力端子からデータ y_{10} が供給される。2入力セクタ305は、入力レジスタ301の出力データ y_{10} を選択する。一方、係数メモリ302~304からそれぞれ t_1, t_2, t_3 が読み出され、乗算器306~308により3個の積 $t, y_{10}, t_1 y_{10}, t_2 y_{10}, t_3 y_{10}$ が並列に計算される。次に、累算器310~317の4入力セクタ201により、一時レジスタ309の出力データと3個の乗算器306~308の結果とのうちの1個がそれぞれ選択される。この場合、第1の累算器310では一時レジスタ309の出力データ t, y_{10} が、第2の累算器311では第1の乗算器306の結果 t, y_{10} が、第3の累算器312では第2の乗算器307の結果 t, y_{10} が、第4の累算器313では第3の乗算器308の結果 t, y_{10} が、第5の累算器314では第3の乗算器308の結果 t, y_{10} が、第6の累算器315では第2の乗算器307の結果

t_1, y_{30} が、第7の累算器316では第1の乗算器306の結果 t, y_{30} が、第8の累算器317では一時レジスタ309の出力データ t, y_{30} がそれぞれ選択される。第1、第5、第6及び第7の累算器310、314、315、316の2の補数器202は、それぞれ4入力セクタ201の出力をそのまま通過させる。第2、第3、第4及び第8の累算器311、312、313、317の2の補数器202は、それぞれ4入力セクタ201の出力の2の補数を出力する。累算器310～317の加算器203は、2の補数器202の結果と累算レジスタ204の出力との和を計算し、その加算結果を累算レジスタ204にそれぞれ書き込む。この結果、第1の累算器310では $t, y_{00} + t_1, y_{10} + t_2, y_{20} + t_3, y_{30}$ が、第2の累算器311では $t, y_{00} + t_1, y_{10} + t_2, y_{20} - t_3, y_{30}$ が、第3の累算器312では $t, y_{00} + t_1, y_{10} - t_2, y_{20} - t_3, y_{30}$ が、第4の累算器313では $t, y_{00} + t_1, y_{10} - t_2, y_{20} - t_3, y_{30}$ が、第5の累算器314では $t, y_{00} - t_1, y_{10} - t_2, y_{20} + t_3, y_{30}$ が、第6の累算器315では $t, y_{00} - t_1, y_{10} - t_2, y_{20} + t_3, y_{30}$ が、第7の累算器316では $t, y_{00} - t_1, y_{10} + t_2, y_{20} + t_3, y_{30}$ が、第8の累算器317では $t, y_{00} - t_1, y_{10} + t_2, y_{20} - t_3, y_{30}$ がそれぞれ累算レジスタ204に格納される。更に、第5のサイクルの終わりで、データ y_{40} が入力レジスタ301に書き込まれ、かつ第1の乗算器306の結果 t, y_{30} が一時レジスタ309に書き込ま*

＊れる。

【0036】第6から第9のサイクルでは、入力端子からデータ $y_{50}, y_{60}, y_{70}, y_{80}$ が順次供給される。したがって、第9サイクルの終りには、累算器310～317の累算レジスタ204に、8個の内積 $w_{00}, w_{10}, w_{20}, w_{30}, w_{40}, w_{50}, w_{60}, w_{70}$ が格納される。更に、第9のサイクルの終わりで、データ y_{80} が入力レジスタ301に書き込まれ、かつ第1の乗算器306の結果 t, y_{70} が一時レジスタ309に書き込まれる。

【0037】第10のサイクルでは、入力端子からデータ y_{90} が供給されて上記第2のサイクルと同様の処理が実行されるとともに、累算器310～317の累算レジスタ204の保持内容 $w_{00}, w_{10}, w_{20}, w_{30}, w_{40}, w_{50}, w_{60}, w_{70}$ がバッファレジスタ205へそれぞれ転送される。そして、8入力セクタ318は、第1の累算器310の出力 w_{00} を選択出力する。

【0038】以下同様の処理を繰り返すことにより、連続的に供給される入力データ $y_{00} \sim y_{70}, y_{80} \sim y_{90}, \dots, y_{67} \sim y_{77}$ に対応した出力データ $w_{00} \sim w_{70}, w_{80} \sim w_{90}, \dots, w_{67} \sim w_{77}$ が連続して得られる。

【0039】以上のとおり、第2の実施例によれば、乗算器の数が3に低減された1次元IDCTプロセッサを実現できる。

【0040】(実施例3) 式(4)から、 $t_0 = t$ 、であることが直ちに分かる。この関係を利用すると、式(2)は、

$$\begin{aligned} w_{i1} &= d_{i0} y_{01} + \sum_{k=1}^7 d_{ik} y_{k1} + d_{i4} y_{41} + \sum_{k=5}^7 d_{ik} y_{k1} \\ &= t_0 y_{01} + \sum_{k=1}^7 d_{ik} y_{k1} \pm t_0 y_{41} + \sum_{k=5}^7 d_{ik} y_{k1} \\ &= t_0 y_{01} \pm t_0 y_{41} + \sum_{k=1}^7 d_{ik} y_{k1} + \sum_{k=5}^7 d_{ik} y_{k1} \\ &= t_0 y_{01} \pm t_0 y_{41} + \omega_{i1} \end{aligned} \quad \dots (5)$$

のように変形される。ここに、式(5)中の“ \pm ”は、 $i=0, 3, 4, 7$ の場合に“+”を、 $i=1, 2, 5, 6$ の場合に“-”をそれぞれ意味する(図1参照)。また、式(5)中の ω_{i1} は部分内積であって、

$$\omega_{i1} = \sum_{k=1}^7 d_{ik} y_{k1} + \sum_{k=5}^7 d_{ik} y_{k1} \quad \dots (6)$$

である。式(6)によれば、図1の行列演算のサイズは、図10のように低減される。

【0041】本発明の第3の実施例に係る1次元IDCTプロセッサの構成を図11に示す。この構成は、式(5)の演算を実行するものである。図11において、10は入力バッファ、11は定数乗算回路、12は分布演算(DA)回路、13は合成演算(RA)回路である。入力端子から、16ビット長の2の補数表示の2進数データ y_{i1} ($i=0 \sim 7, j=0 \sim 7$)が、 $y_{00} \sim y_{70}, y_{80} \sim y_{90}, \dots, y_{67} \sim y_{77}$ の順序で入力バッファ10へ供給される。入力バッファ10は、データ y_{01}, y_{41} を定数乗算回路11へ、データ $y_{11}, y_{21}, y_{31}, y_{51}, y_{61}, y_{71}$ をDA回路12へそれぞれ供給する。定数乗算回路11は、2個の定数乗算 $t_0 y_{01}, t_0 y_{41}$ を実行するものである。DA回路12は、図10の行列演算を実行することにより部分内積 ω_{i1} を求めるものである。RA回路13は、 $t_0 y_{01}, t_0 y_{41}$ 及び ω_{i1}

※5、6の場合に“-”をそれぞれ意味する(図1参照)。また、式(5)中の ω_{i1} は部分内積であって、

から、式(5)に従って内積 w_{i1} を求めるものである。【0042】入力バッファ10の内部構成を図12に示す。入力バッファ10は、各々データ $y_{01}, y_{11}, y_{21}, y_{31}, y_{41}, y_{51}, y_{61}, y_{71}$ を保持するための8個のレジスタ400～407で構成される。

【0043】定数乗算回路11の内部構成を図13に示す。定数乗算回路11は、データ y_{01} を保持するための入力レジスタ410と、データ y_{41} を保持するための入力レジスタ411と、2個のデータ y_{01}, y_{41} を順次選択するための2入力セクタ412と、2個の定数乗算 $t_0 y_{01}, t_0 y_{41}$ を順次実行するための乗算器413と、積 $t_0 y_{01}$ を保持するための一時レジスタ414と、積 $t_0 y_{41}$ を保持するための一時レジスタ415と、当該1次元IDCTプロセッサのパイプライン動作を保証するように両一時レジスタ414、415の出力を保持するための2個のバッファレジスタ416、417とで構成される。

【0044】DA回路12の内部構成を図14に示す。DA回路12は、6個のシフトレジスタ420~425と、8個の6ビット入力RAC426~433と、8個のバッファレジスタ434~441と、8入力セレクタ442とで構成される。シフトレジスタ420~425は、各々データ y_{21} , y_{21} , y_{31} , y_{51} , y_{61} , y_{71} を保持し、各々の最下位2ビットを次々とシフトアウトするものである。シフトレジスタ420~425の各々の最下位ビットは第1のビットスライスワード q_0 として、各々の最下位ビットより1桁上位のビットは第2のビットスライスワード q_1 としてそれぞれ6ビット入力RAC426~433へ供給される。6ビット入力RAC426は、図15に示すように、第1のROM71と、第2のROM72と、3入力加減算器73と、シフタ74と、累算レジスタ75とで構成される。第1のROM71は、第1のビットスライスワード q_0 をアドレスとして受け取り、対応するベクトル内積の部分積を3入力加減算器73へ第1の入力として供給するものである。第2のROM72は、第2のビットスライスワード q_1 をアドレスとして受け取り、対応するベクトル内積の部分積を3入力加減算器73へ第2の入力として供給するものである。累算レジスタ75の保持出力は、3入力加減算器73へ第3の入力として供給される。ただし、第2の入力は、第1及び第3の入力より1ビット上位の重みを持つ。累算レジスタ75の保持内容は、予め0に初期化される。3入力加減算器73は、第1~第3の入力の加算を実行するものである。ただし、最後のビットスライスワード q_1 に係る部分積については、減算を実行する。シフタ74は、3入力加減算器73の結果の桁移動のための左シフタである。累算レジスタ75の保持内容は、シフタ74の出力に書き換えられる。最終的に、累算レジスタ75から部分内積 ω_{01} が出力される。図14中の他の6ビット入力RACの内部構成も図15と同様である。したがって、8個の6ビット入力RAC426~433で8個の部分内積 ω_{01} , ω_{11} , ω_{21} , ω_{31} , ω_{41} , ω_{51} , ω_{61} , ω_{71} が並列に求められる。バッファレジスタ434~441は、当該1次元IDCTプロセッサのパイプライン動作を保證するように、6ビット入力RAC426~433の出力を保持するものである。8入力セレクタ442は、バッファレジスタ434~441の保持データを順次選択して、部分内積 ω_{ij} ($i=0\sim7$, $j=0\sim7$)を、 ω_{01} , ω_{11} , ω_{21} , ω_{31} , ω_{41} , ω_{51} , ω_{61} , ω_{71} の順序で出力するものである。

【0045】RA回路13の内部構成を図16に示す。RA回路13は、定数乗算回路11から供給された2個の積 t , y_{01} , t , y_{41} と、DA回路12から供給された部分内積 ω_{11} との加減算を実行して内積 w_{11} を求めるための3入力加減算器450で構成される。ただし、積 t , y_{41} については、式(5)に従って、 i の値に応じ

て加算又は減算が選択される。具体的には、 $i=0$, 3, 4, 7のサイクルでは加算を選択し、 $i=1$, 2, 5, 6のサイクルでは減算を選択するように制御される。

【0046】以下、図11~図16に基づいて、本発明の第3の実施例に係る1次元IDCTプロセッサの動作を説明する。

【0047】第1から第8のサイクルでは、入力端子から入力バッファ10に8個のデータ y_{00} , y_{10} , y_{20} , y_{30} , y_{40} , y_{50} , y_{60} , y_{70} が順次入力される。これらのデータは、それぞれレジスタ400~407に格納される。

【0048】第9のサイクルでは、入力バッファ10のデータが定数乗算回路11及びDA回路12へ転送される。すなわち、データ y_{00} , y_{40} は定数乗算回路11の入力レジスタ410, 411に、データ y_{10} , y_{20} , y_{30} , y_{50} , y_{60} , y_{70} はDA回路12のシフトレジスタ420~425にそれぞれ格納される。

【0049】第10から第13のサイクルでは、定数乗算回路11の2入力セレクタ412によりデータ y_{00} が選択され、乗算器413により定数乗算 t , y_{00} が実行され、その結果が一時レジスタ414に書き込まれる。第14から第17のサイクルでは、2入力セレクタ412によりデータ y_{40} が選択され、乗算器413により定数乗算 t , y_{40} が実行され、その結果が一時レジスタ415に書き込まれる。一方、DA回路12では、第10から第17のサイクルにおいて、6ビット入力RAC426~433により8個の部分内積 ω_{00} , ω_{10} , ω_{20} , ω_{30} , ω_{40} , ω_{50} , ω_{60} , ω_{70} が求められる。

【0050】第18のサイクルでは、定数乗算回路11の一時レジスタ414, 415の保持データがバッファレジスタ416, 417へ、DA回路12の6ビット入力RAC426~433の出力データがバッファレジスタ434~441へそれぞれ転送される。

【0051】第19から第26のサイクルでは、DA回路12の8入力セレクタ442が部分内積 ω_{00} , ω_{10} , ω_{20} , ω_{30} , ω_{40} , ω_{50} , ω_{60} , ω_{70} をRA回路13へ順次供給する。一方、積 t , y_{00} , t , y_{40} が定数演算回路11からRA回路13へ供給される。RA回路11の3入力加減算器450は、式(5)に従って、内積 w_{00} , w_{10} , w_{20} , w_{30} , w_{40} , w_{50} , w_{60} , w_{70} を順次出力する。

【0052】次の8個のデータ y_{01} , y_{11} , y_{21} , y_{31} , y_{41} , y_{51} , y_{61} , y_{71} に関する処理は、第9から第34のサイクルにおいて、上記第1から第26のサイクルの処理と同様に行われる。この結果、第27から第34のサイクルで内積 w_{01} , w_{11} , w_{21} , w_{31} , w_{41} , w_{51} , w_{61} , w_{71} が順次出力される。

【0053】以下同様の処理を繰り返すことにより、連続的に供給される入力データ $y_{00}\sim y_{70}$, $y_{01}\sim y_{71}$,

..., $y_{0,7} \sim y_{7,7}$ に対応した出力データ $w_{0,0} \sim w_{7,0}$, $w_{0,1} \sim w_{7,1}$, ..., $w_{0,7} \sim w_{7,7}$ が連続して得られる。

【0054】以上のとおり、第3の実施例によれば、乗算器の数が1に低減された1次元IDCTプロセッサを実現できる。しかも、定数乗算回路11の中の乗算器4*

$$w_{i,1} = t_i \cdot (y_{0,i} \pm y_{4,i}) + \omega_{i,1}$$

のように変形される。ここに、式(7)中の“±”は、 $i=0, 3, 4, 7$ の場合に“+”を、 $i=1, 2, 5, 6$ の場合に“-”をそれぞれ意味する(図1参照)。図17の定数乗算回路11及び図18のRA回路13は、式(7)の演算手順を採用したものである。

【0056】図17に示した定数乗算回路11は、データ $y_{0,i}$ を保持するための入力レジスタ500と、データ $y_{4,i}$ を保持するための入力レジスタ501と、加算 $y_{0,i} + y_{4,i}$ 及び減算 $y_{0,i} - y_{4,i}$ を順次実行するための2入力加減算器502と、2個の定数乗算 $t_i \cdot (y_{0,i} + y_{4,i})$, $t_i \cdot (y_{0,i} - y_{4,i})$ を順次実行するための乗算器503と、積 $t_i \cdot (y_{0,i} + y_{4,i})$ を保持するための一時レジスタ504と、積 $t_i \cdot (y_{0,i} - y_{4,i})$ を保持するための一時レジスタ505と、両一時レジスタ504, 505の出力を保持するための2個のバッファレジスタ506, 507とで構成される。

【0057】図17の定数乗算回路11を採用する場合には、図16のRA回路13は図18のように変形される。図18のRA回路13は、定数乗算回路11から供*

$$\omega_{0,1} = \rho_{0,1} + \sigma_{0,1}$$

$$\omega_{1,1} = \rho_{1,1} + \sigma_{1,1}$$

$$\omega_{2,1} = \rho_{2,1} + \sigma_{2,1}$$

$$\omega_{3,1} = \rho_{3,1} + \sigma_{3,1}$$

$$\omega_{4,1} = \rho_{3,1} - \sigma_{3,1}$$

$$\omega_{5,1} = \rho_{2,1} - \sigma_{2,1}$$

$$\omega_{6,1} = \rho_{1,1} - \sigma_{1,1}$$

$$\omega_{7,1} = \rho_{0,1} - \sigma_{0,1}$$

であることが分かる。図21のDA回路12は、図20(a)及び図20(b)の行列演算をそれぞれRACで実行したうえ、式(8)を用いて部分内積 $\omega_{i,1}$ を求めるものである。

【0059】図21のDA回路12は、6個のシフトレジスタ700~705と、4個の4ビット入力RAC706~709と、4個の2ビット入力RAC710~713と、8個のバッファレジスタ714~721と、第1の4入力セクタ722と、第2の4入力セクタ723と、2入力加減算器724とで構成される。シフトレジスタ700~705は、各々データ $y_{1,1}$, $y_{2,1}$, $y_{3,1}$, $y_{4,1}$, $y_{5,1}$, $y_{6,1}$ を保持し、各々の最下位2ビットを次々とシフトアウトするものである。4個のシフトレジスタ700, 702, 703, 705の各々の最下位ビットは第1のビットスライズワード s_0 として、各々の最下位ビットより1桁上位のビットは第2のビットスライズワード s_1 としてそれぞれ4ビット入力RAC7

*13は、2変数入力の乗算器に比べて回路規模が小さい。また、内積計算の一部を定数乗算回路11で実行するので、DA回路12のROMサイズが低減される。

【0055】以下、上記第3の実施例の変形例について説明する。式(5)は、

$$\dots (7)$$

※給された2個の積 $t_i \cdot (y_{0,i} + y_{4,i})$, $t_i \cdot (y_{0,i} - y_{4,i})$ のうちのいずれか一方を選択するための2入力セクタ510と、該2入力セクタ510で選択された積とDA回路12から供給された部分内積 $\omega_{i,1}$ との加算を実行して内積 $w_{i,1}$ を求めるための2入力加算器511とで構成される。2入力セクタ510は、式(7)に従って、 $i=0, 3, 4, 7$ のサイクルでは $t_i \cdot (y_{0,i} + y_{4,i})$ を選択し、 $i=1, 2, 5, 6$ のサイクルでは $t_i \cdot (y_{0,i} - y_{4,i})$ を選択するように制御される。

【0058】さて、図10の行列演算は、図19のように変形される。図19中の8行6列の行列の要素の半分は0である。したがって、図19の行列演算は、図20(a)及び図20(b)のように2つに分割される。図20(a)中の4個の部分内積 $\rho_{0,1}$, $\rho_{1,1}$, $\rho_{2,1}$, $\rho_{3,1}$ は4個の2ビット入力RACで、図20(b)中の4個の部分内積 $\sigma_{0,1}$, $\sigma_{1,1}$, $\sigma_{2,1}$, $\sigma_{3,1}$ は4個の4ビット入力RACでそれぞれ求めることができる。また、図20(a)及び図20(b)から、

$$\dots (8)$$

06~709へ供給される。2個のシフトレジスタ701, 704の各々の最下位ビットは第3のビットスライズワード r_1 として、各々の最下位ビットより1桁上位のビットは第4のビットスライズワード r_2 としてそれぞれ2ビット入力RAC710~713へ供給される。4ビット入力RAC706は、図22に示すように、第1のROM81と、第2のROM82と、3入力加減算器83と、シフタ84と、累算レジスタ85とで構成される。図21中の他の4ビット入力RACの内部構成も図22と同様である。したがって、4個の4ビット入力RAC706~709で4個の部分内積 $\sigma_{0,1}$, $\sigma_{1,1}$, $\sigma_{2,1}$, $\sigma_{3,1}$ が並列に求められる。2ビット入力RAC710は、図23に示すように、第1のROM91と、第2のROM92と、3入力加減算器93と、シフタ94と、累算レジスタ95とで構成される。図21中の他の2ビット入力RACの内部構成も図23と同様である。したがって、4個の2ビット入力RAC710~713

で4個の部分内積 ρ_{01} , ρ_{11} , ρ_{21} , ρ_{31} が並列に求められる。バッファレジスタ714~721は、当該1次元IDCTプロセッサのパイプライン動作を保證するように、8個のRAC706~713の出力を保持するものである。第1の4入力セクタ722は、バッファレジスタ714~717の保持データを選択して、部分内積 σ_{01} , σ_{11} , σ_{21} , σ_{31} , σ_{41} , σ_{51} , σ_{61} , σ_{71} を2入力加減算器724へ順次供給するものである。第2の4入力セクタ723は、バッファレジスタ718~721の保持データを選択して、部分内積 ρ_{01} , ρ_{11} , ρ_{21} , ρ_{31} , ρ_{41} , ρ_{51} , ρ_{61} , ρ_{71} を2入力加減算器724へ順次供給するものである。2入力加減算器724は、式(8)に従って加減算を実行するものである。すなわち、部分内積 ω_{ij} ($i=0\sim7$, $j=0\sim7$)が、 ω_{01} , ω_{11} , ω_{21} , ω_{31} , ω_{41} , ω_{51} , ω_{61} , ω_{71} の順序で2入力加減算器724から出力される。

【0060】図11中の定数演算回路11及びRA回路13の内部構成は、図13と図16との組み合わせ、図17と図18との組み合わせなどの中から適宜選択される。また、図11中のDA回路12の内部構成は、図14及び図21などの中から適宜選択される。

【0061】なお、上記第1~第3の実施例では8ポイントIDCT処理について説明したが、各実施例は16ポイントIDCT処理、8ポイントIDST処理、16ポイントIDST処理などに容易に変形できる。

【0062】

【発明の効果】以上説明してきたとおり、本発明によれば、所要の乗算器数が大幅に低減される結果、直交変換プロセッサの回路規模が低減される。また、複数の内積計算の各々を2個の定数乗算と1個の部分内積計算とに分割することとすれば、内積計算の全てをDA回路で実現する場合に比べてROMサイズが低減される結果、直交変換プロセッサの回路規模が低減される。

【図面の簡単な説明】

【図1】本発明に係るIDCTプロセッサによって実行されるべき行列演算を示す図である。

【図2】図1の行列演算の1つの実行手順を示す図である。

【図3】図2の手順で用いられる係数行列を示す図である。

【図4】図3の係数行列を採用した、本発明の第1の実施例に係るIDCTプロセッサの構成図である。

【図5】図4中の1個の累算器の内部構成図である。

【図6】図5の累算器の変形例を示す図である。

【図7】図1の行列演算の他の実行手順を示す図である。

【図8】図7の手順で用いられる係数行列を示す図である。

【図9】図8の係数行列を採用した、本発明の第2の実施例に係るIDCTプロセッサの構成図である。

【図10】図1の行列演算の一部を示す図である。

【図11】本発明の第3の実施例に係るIDCTプロセッサの構成図である。

【図12】図11中の入力バッファの内部構成図である。

【図13】図11中の定数乗算回路の内部構成図である。

【図14】図11中の分布演算回路の内部構成図である。

【図15】図10の行列演算を実行するための、図14中の1個の6ビット入力RACの内部構成図である。

【図16】図11中の合成演算回路の内部構成図である。

【図17】図13の定数乗算回路の変形例を示す図である。

【図18】図17の定数乗算回路を採用したIDCTプロセッサにおける合成演算回路の内部構成図である。

【図19】図10から導出された行列演算を示す図である。

【図20】(a)及び(b)は図19から分割された2つの行列演算を示す図である。

【図21】図14の分布演算回路の変形例を示す図である。

【図22】図20(b)の行列演算を実行するための、図21中の1個の4ビット入力RACの内部構成図である。

【図23】図20(a)の行列演算を実行するための、図21中の1個の2ビット入力RACの内部構成図である。

【符号の説明】

10 入力バッファ

11 定数乗算回路

12 分布演算回路(DA回路)

13 合成演算回路(RA回路)

71, 72, 81, 82, 91, 92 ROM

73, 83, 93 3入力加減算器

74, 84, 94 シフタ

75, 85, 95 累算レジスタ

101~104, 302~304 係数メモリ

40 105~108, 306~308 乗算器

109~116, 310~317 累算器

117, 318 8入力セクタ

201 4入力セクタ

202 2の補数器

203 加算器

204 累算レジスタ

205 バッファレジスタ

212 1の補数器

301 入力レジスタ

50 305 2入力セクタ

309 一時レジスタ

400~407 レジスタ

410, 411, 500, 501 入力レジスタ

412, 510 2入力セレクタ

413, 503 乗算器

414, 415, 504, 505 一時レジスタ

416, 417, 506, 507 バッファレジスタ

420~425, 700~705 シフトレジスタ

426~433 6ビット入力RAC

* 434~441, 714~721 バッファレジスタ

442 8入力セレクタ

450 3入力加減算器

502 2入力加減算器

511 2入力加算器

706~709 4ビット入力RAC

710~713 2ビット入力RAC

722, 723 4入力セレクタ

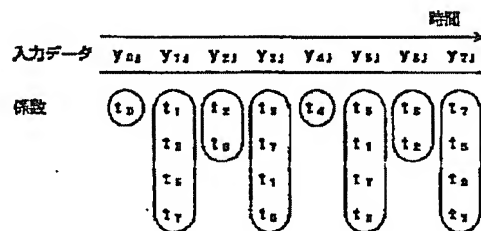
* 724 2入力加減算器

【図1】

$$\begin{bmatrix} W_{0j} \\ W_{1j} \\ W_{2j} \\ W_{3j} \\ W_{4j} \\ W_{5j} \\ W_{6j} \\ W_{7j} \end{bmatrix} = \begin{bmatrix} t_0 & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \\ t_0 & t_2 & t_6 & -t_7 & -t_4 & -t_1 & -t_2 & -t_5 \\ t_0 & t_5 & -t_6 & -t_1 & -t_4 & t_7 & t_2 & t_0 \\ t_0 & t_7 & -t_2 & -t_5 & t_4 & t_2 & -t_6 & -t_1 \\ t_0 & -t_7 & -t_2 & t_5 & t_4 & -t_2 & -t_6 & t_1 \\ t_0 & -t_5 & -t_1 & t_1 & -t_4 & -t_7 & t_2 & -t_5 \\ t_0 & -t_2 & t_5 & t_7 & -t_4 & t_1 & -t_2 & t_0 \\ t_0 & -t_1 & t_2 & -t_3 & t_4 & -t_5 & t_6 & -t_7 \end{bmatrix} \begin{bmatrix} Y_{0j} \\ Y_{1j} \\ Y_{2j} \\ Y_{3j} \\ Y_{4j} \\ Y_{5j} \\ Y_{6j} \\ Y_{7j} \end{bmatrix}$$

【図2】

【図3】

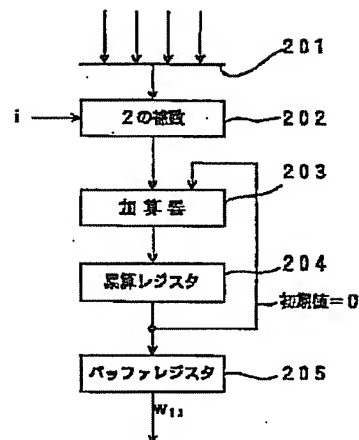
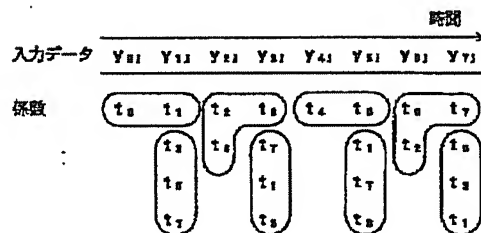


【図7】

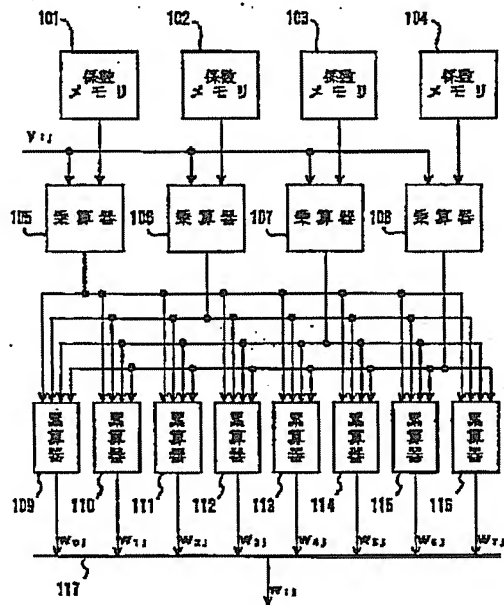
$$E = \begin{bmatrix} t_0 & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \\ t_0 & t_3 & t_2 & t_7 & t_4 & t_1 & t_2 & t_5 \\ t_0 & t_2 & t_5 & t_1 & t_4 & t_7 & t_2 & t_0 \\ t_0 & t_7 & t_2 & t_5 & t_4 & t_3 & t_5 & t_1 \end{bmatrix}$$

【図5】

110

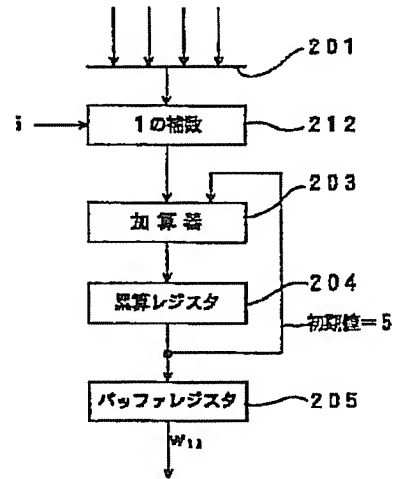


【図4】



【図6】

110

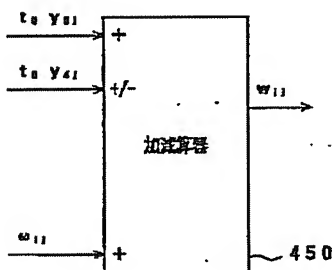


【図8】

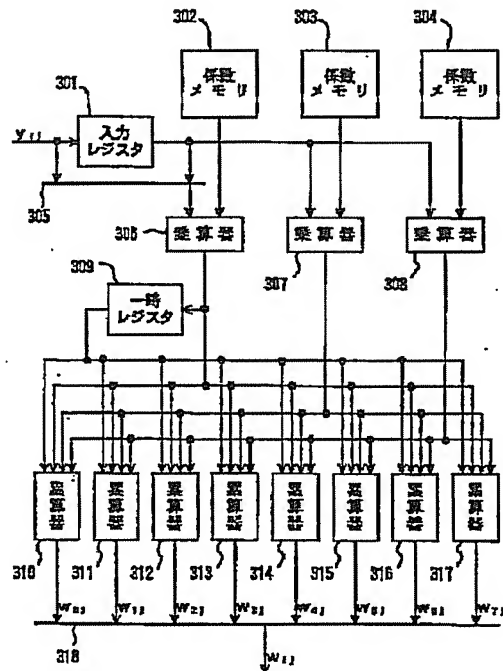
$$G = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \\ t_2 & t_1 & t_2 & t_1 & t_4 & t_7 & t_8 & t_2 \\ t_8 & t_7 & t_8 & t_8 & t_4 & t_2 & t_8 & t_1 \end{bmatrix}$$

【図16】

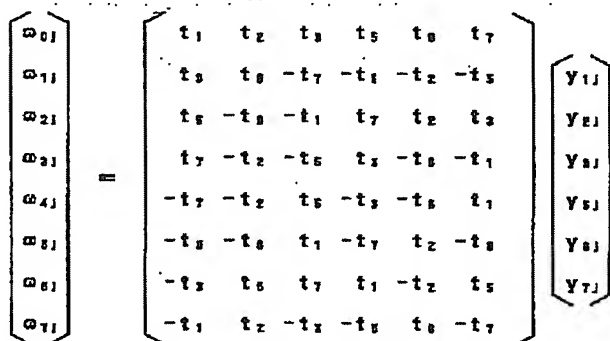
13



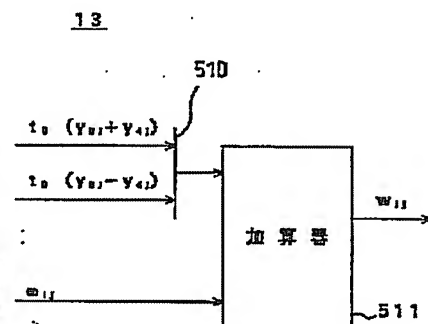
【図9】



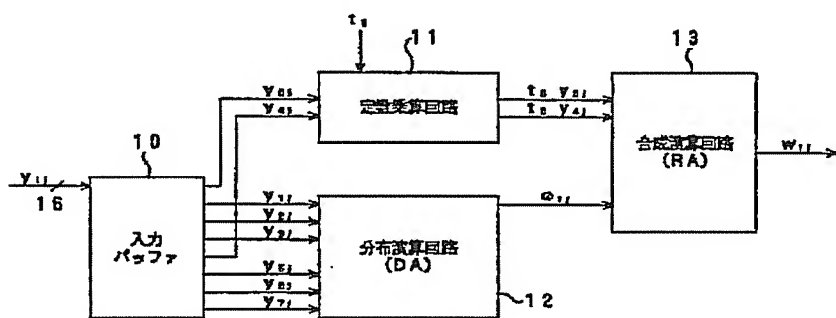
【図10】



【図18】

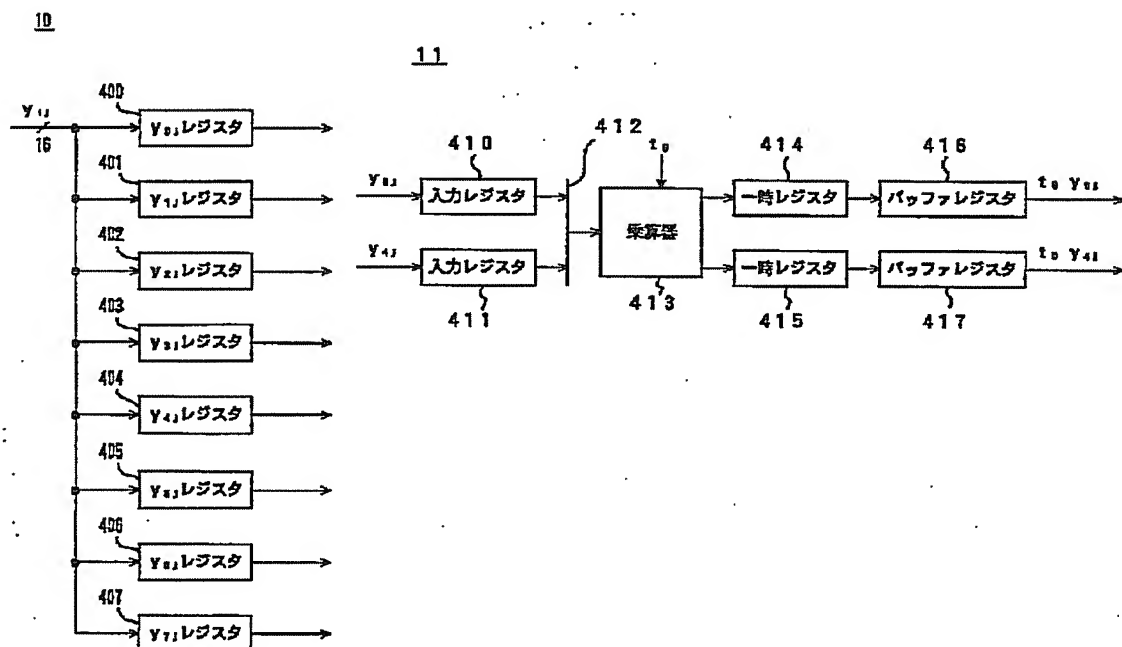


【図11】

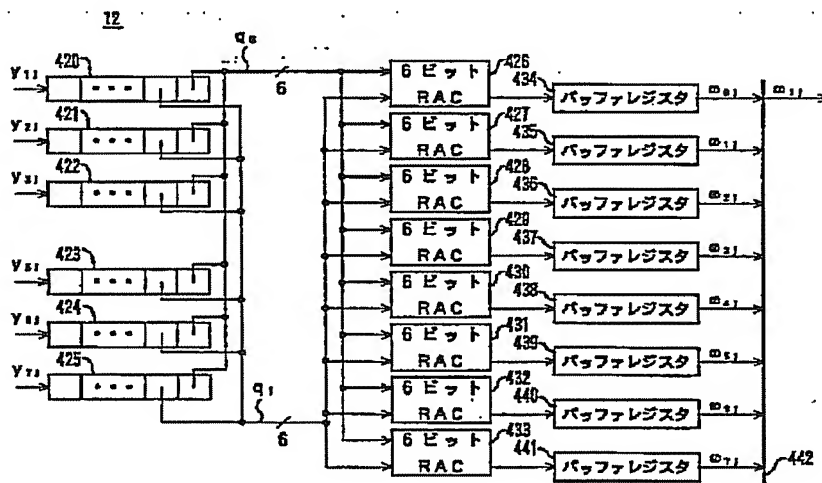


【図12】

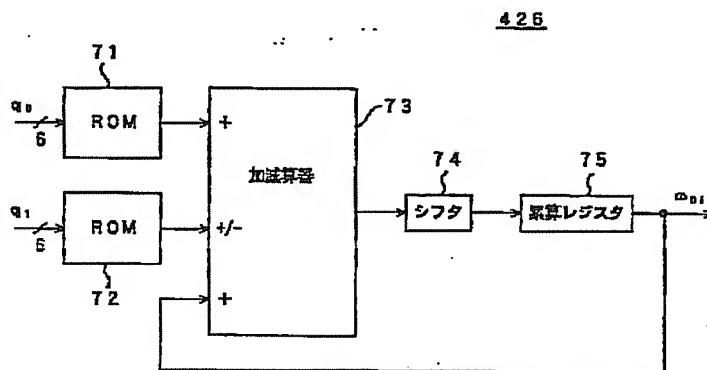
【図13】



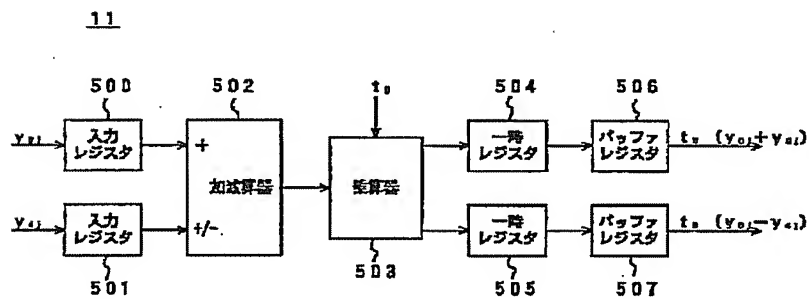
【図14】



【図15】



【図17】



【図19】

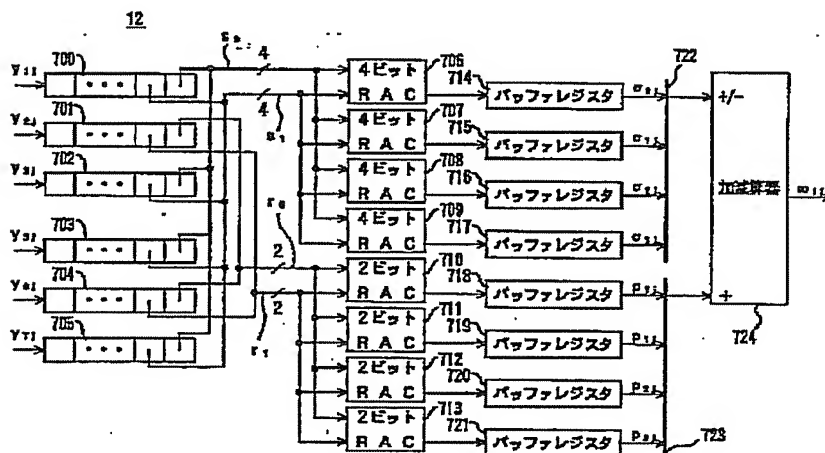
$$\begin{bmatrix} \omega_{0j} + \omega_{7j} \\ \omega_{1j} + \omega_{6j} \\ \omega_{2j} + \omega_{5j} \\ \omega_{3j} + \omega_{4j} \\ \omega_{0j} - \omega_{7j} \\ \omega_{1j} - \omega_{6j} \\ \omega_{2j} - \omega_{5j} \\ \omega_{3j} - \omega_{4j} \end{bmatrix} = 2 \begin{bmatrix} 0 & t_z & 0 & 0 & t_0 & 0 \\ 0 & t_0 & 0 & 0 & -t_z & 0 \\ 0 & -t_0 & 0 & 0 & t_z & 0 \\ 0 & -t_z & 0 & 0 & -t_0 & 0 \\ t_1 & 0 & t_5 & -t_3 & 0 & t_1 \\ t_0 & 0 & t_1 & -t_7 & 0 & -t_5 \\ t_5 & 0 & t_7 & t_1 & 0 & t_5 \\ t_7 & 0 & -t_5 & -t_5 & 0 & -t_7 \end{bmatrix} \begin{bmatrix} Y_{1j} \\ Y_{2j} \\ Y_{3j} \\ Y_{5j} \\ Y_{6j} \\ Y_{7j} \end{bmatrix}$$

【図20】

$$(a) \quad \frac{1}{2} \begin{bmatrix} \omega_{0j} + \omega_{7j} \\ \omega_{1j} + \omega_{6j} \\ \omega_{2j} + \omega_{5j} \\ \omega_{3j} + \omega_{4j} \end{bmatrix} = \begin{bmatrix} t_z & t_0 \\ t_0 & -t_z \\ -t_0 & t_z \\ -t_z & -t_0 \end{bmatrix} \begin{bmatrix} Y_{2j} \\ Y_{6j} \end{bmatrix} = \begin{bmatrix} P_{0j} \\ P_{1j} \\ P_{2j} \\ P_{3j} \end{bmatrix}$$

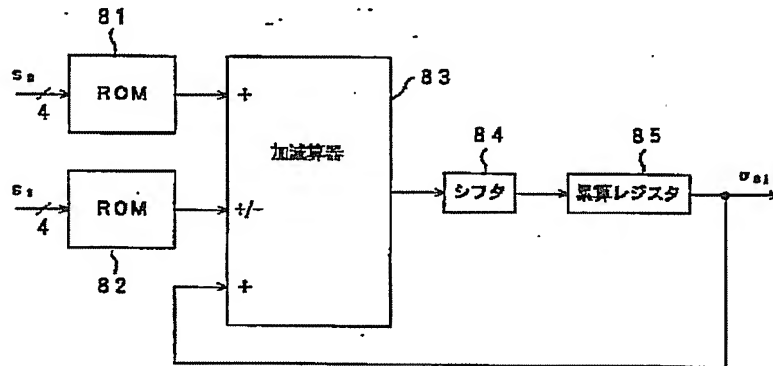
$$(b) \quad \frac{1}{2} \begin{bmatrix} \omega_{0j} - \omega_{7j} \\ \omega_{1j} - \omega_{6j} \\ \omega_{2j} - \omega_{5j} \\ \omega_{3j} - \omega_{4j} \end{bmatrix} = \begin{bmatrix} t_1 & t_5 & -t_3 & t_1 \\ t_5 & t_1 & -t_7 & -t_5 \\ t_5 & t_7 & t_1 & t_5 \\ t_7 & -t_5 & -t_5 & -t_7 \end{bmatrix} \begin{bmatrix} Y_{1j} \\ Y_{3j} \\ Y_{5j} \\ Y_{7j} \end{bmatrix} = \begin{bmatrix} \sigma_{1j} \\ \sigma_{2j} \\ \sigma_{3j} \\ \sigma_{4j} \end{bmatrix}$$

【図21】



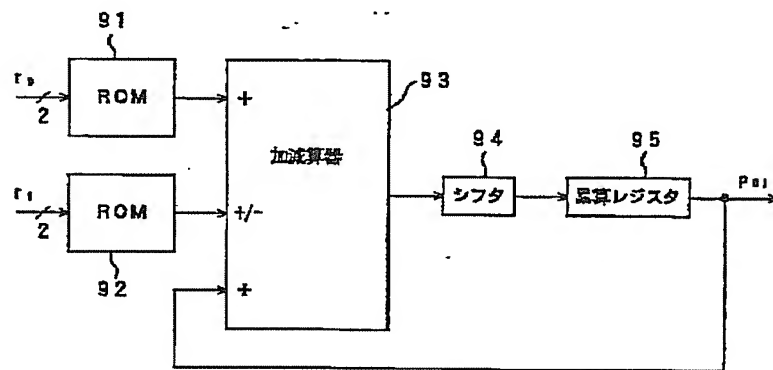
【図22】

706



【図23】

710



フロントページの続き

(51)Int.Cl.⁶

H03M 7/30

H04N 1/41

識別記号

片内整理番号

FI

技術表示箇所

A 9382-5K

B